



<https://verifone.cloud/docs/online-payments/checkout/accepting-payments/op-online-payments>

Updated: 12-Nov-2024

## OP Online payments

### Overview

Checkout can be used to accept Op Online bank payments.

Integration options supported: Hosted payment page (HPP) or IFRAME.

This guide requires familiarity with [Accepting payments](#).

## Integrate Op Online Payments via Checkout

| Parameters                | Type    | Required | Description  |
|---------------------------|---------|----------|--|
| currency_code             | String  | Yes      | Supported currencies: EUR  |
| amount                    | Integer | Yes      | Transaction amount   |
| customer                  | String  | No       | ID of a Customer created via the <a href="#">Customer API</a> . A customer object can be created and attached to a Checkout                              |
| configurations.bank       | Object  | Yes      | Object carrying the parameters required for making a Bank payment  |
| dynamic_descriptor        | String  | No       | Short text to be displayed on the bank statement of the cardholder. Support varies per Payment Contract  |
| authorization_type        | String  | No       | Authorization Type (FINAL_AUTH)  |
| shopper_interaction       | String  | No       | Enum: "ECOMMERCE" "MAIL". Indicates the channels used by the shopper to send the payment data for transactions   |
| configurations.bank.op    | Object  | No       | Object carrying the parameters required for OP Bank payments   |
| payment_contract_id       | String  | Yes      | This ID can be found in the portal or given to you by a Verifone employee. It is used to retrieve a merchant's OP bank credentials necessary for payment |
| multi_bank_payment_option | Boolean | No       | Indicator for using OP partner banks (example: Nordea, Danske Bank, S-Pankki and others)   |

### Authorization and capture

You can use Op Online Payment transactions through the Checkout to do a sale transaction (authorization + capture in the same step).

### **Sending a checkout request using Op Online Payment method:**

```
{
  "currency_code": "EUR",
  "entity_id": "466c5c59-5177-45c2-b9a7-41ac422ef1fa",
  "customer": "deebacec-cd75-4ce5-9641-bfc40de5710d",
  "configurations": {
    "bank": {
      "dynamic_descriptor": "Test Product",

      "op" :{
        "payment_contract_id":
"98a003be-03f2-43c4-a89c-36021fa63635",

      }
    },
    "amount": 4300,
    "merchant_reference": "Test",
    "return_url": "http://enbyhy7yz2lg98r.m.pipedream.net",
    "interaction_type": "HPP"
  }
}
```

### **Handling responses**

Whenever a bank to bank transaction is processed via the Checkout, the responses events include the following parameters:

Example of a successful OP Online payment via the Checkout:

```
[
  {
    "type": "SUBMITTED",
    "id": "2207e3b4-3e6d-431f-82b3-08bdfd6f8f1b",
    "timestamp": "2024-04-10T12:18:15.404Z"
  },
  {
    "type": "TRANSACTION_INITIATED",
    "id": "f06259a4-f58c-4ece-8c99-1be4b3759732",
    "timestamp": "2024-04-10T12:18:16.394Z",
    "details": {
      "amount": 200,
      "status": "INITIATED",
      "id": "549b2e15-eb33-44eb-ac78-c66fdc7bae6b",
      "payment_url":
"https://api.smn-sandbox.aws.op-palvelut.fi/customer/payment/multibank/direct/Ban
",
      "processor": "OP_ONLINE_PAYMENT",
      "payment_product": "OP_ONLINE_PAYMENT"
    }
  }
]
```

```

    }
  },
  {
    "type": "TRANSACTION_SUCCESS",
    "id": "9a6c3ffe-ce56-4cf5-8879-0f7ec2a31a84",
    "timestamp": "2024-04-10T12:18:17.743Z",
    "details": {
      "id": "549b2e15-eb33-44eb-ac78-c66fdc7bae6b",
      "processor_reference": null,
      "status": "SETTLED",
      "authorization_code": null,
      "cvv_result": null,
      "details": {
        "mid": "230729"
      },
      "reason_code": null,
      "rrn": null,
      "stan": null,
      "additional_data": {
        "acquirer_response_code": null,
        "initiator_trace_id": null
      }
    }
  }
]

```

#### Example of a failed OP Online payment via the Checkout:

```

[
  {
    "type": "SUBMITTED",
    "id": "9b28e580-14e4-473d-b404-c5a6b05b4930",
    "timestamp": "2024-04-10T12:19:53.760Z"
  },
  {
    "type": "TRANSACTION_FAILED",
    "id": "6a2e3da8-97fc-432b-a21b-24053bdfb1ba",
    "timestamp": "2024-04-10T12:19:53.863Z",
    "details": {
      "error": {
        "code": 400,
        "details": {
          "currencyCode":
"must be one of the following values: EUR"
        },
        "timestamp": 1712751593845,
        "message":
"At least one parameter is invalid. Examine the details property for more information"
      }
    },
    "operation": "CREATE_ECOM_ORDER",
    "service": "ECOM"
  }
],
{
  "type": "TRANSACTION_FAILED",

```

```

    "id": "96d44ef8-74f6-4547-b53d-f4b3d9350b95",
    "timestamp": "2024-04-10T12:19:53.867Z",
    "details": {
      "errors": [],
      "service": "CO",
      "code": 123,
      "details": {
        "error": {
          "code": 400,
          "details": {
            "currencyCode":
"must be one of the following values: EUR"
          },
          "timestamp": 1712751593845,
          "message":
"At least one parameter is invalid. Examine the details property for more information"
        },
        "operation": "CREATE_ECOM_ORDER",
        "service": "ECOM"
      },
      "name": "TRANSACTION_FAILED_ERROR",
      "status": 400,
      "timestamp": 1712751593862,
      "traceId": null
    }
  }
]

```

To ensure that the redirection request was not tampered with, always check that the `transaction_id` received as query parameter in the redirection matches the `transaction_id` property of the retrieved Checkout. If those are not matching, this is indication of either an incorrect integration, that the redirection to your `return_url` did not originate from Verifone, or `transaction_id` was tampered with.

You can now store the `transaction_id` value together with the order 1234 in your system to link the two together.

## Scenarios

The table below describes the different outcomes of a Checkout. A full list of [error codes](#) is available.

| Description  | Result   | Merchant action  |
|--|--|--|
| Failed transaction*                                      | Redirect: <code>checkout_id={checkout_id}</code> & <code>transaction_id={transaction_id}</code> & <code>errorCode=123</code> | Unsuccessful payment (technical reason), do not display order confirmation   |
| Successful transaction                                   | Redirect: <code>checkout_id={checkout_id}</code> & <code>transaction_id={transaction_id}</code>                              | Display order confirmation   |
| Customer visits the URL of an already completed Checkout | Redirect: <code>checkout_id={checkout_id}</code> & <code>errorCode=168</code>  | Display corresponding message to the customer. Checkout is completed whenever there was a single successful payment processed through it |

| Description  | Result  | Merchant action   |
|--|---|---|
| Customer visits the URL of an expired Checkout   | Redirect: checkout_id={checkout_id} & errorCode=169 | Display corresponding message to the customer. Checkout is expired whenever the expiry_time is reached                          |
| Customer visits the URL of a Checkout which has reached the maximum of failed payment attempts | Redirect: checkout_id={checkout_id} & errorCode=165 | Display corresponding message to customer. Payments through a single Checkout can be attempted up to three times unsuccessfully |
| Form validation errors / Other service failures on the Checkout page                           | Displays error alert to Customer on the page        | Customer is prompted to correct their form input and retry the payment or try using alternate card or payment method            |

\*\*\* Failed transaction - Depending on which step in the payment process failed, the transaction\_id might not always be present as the query parameter